Aplicações de Aprendizado por Reforço e Machine Learning em Estratégias de Trading de Bitcoin

Davi Mattos

Instituto de Informática e Eletrônica

28 de Junho 2025

Abstract

Este trabalho explora o uso de algoritmos de aprendizado por reforço (Reinforcement Learning - RL) e técnicas de aprendizado de máquina (Machine Learning - ML) aplicadas ao desenvolvimento de estratégias automatizadas para o trading de Bitcoin (BTC). O ambiente altamente volátil e não-estacionário dos mercados de criptoativos representa um desafio único para a modelagem preditiva e a tomada de decisão baseada em dados. O estudo apresenta uma arquitetura de pipeline que integra extração e transformação de dados de mercado, engenharia de features financeiras, modelagem supervisionada com redes neurais profundas, e controle sequencial com métodos de aprendizado por reforço como Deep Q-Networks (DQN) e Proximal Policy Optimization (PPO). Avaliamos o desempenho das estratégias através de métricas de retorno ajustado ao risco (Sharpe Ratio, Sortino Ratio) e benchmarks em ambientes simulados e dados históricos. Os resultados mostram que agentes baseados em RL conseguem capturar padrões dinâmicos de preço com desempenho superior ao buy-and-hold, ainda que altamente sensíveis ao tuning de hiperparâmetros e à estrutura de recompensas. Este paper contribui com uma abordagem sistemática e replicável para o uso de RL e ML no contexto de criptoativos, destacando limitações práticas e direções futuras de pesquisa.

Sumário

1. Introdução

   1.1. Motivação

   1.2. Objetivos

   1.3. Organização do Paper

2. Fundamentos Teóricos

   2.1. Bitcoin e Mercados Criptoativos

   2.2. Aprendizado de Máquina em Finanças

   2.3. Aprendizado por Reforço: Fundamentos e Algoritmos

   2.4. Características dos Mercados Voláteis e Não Estacionários

3. Arquitetura da Solução Proposta

   3.1. Coleta e Pré-processamento de Dados

   3.2. Engenharia de Features

   3.3. Modelos Supervisionados: Previsão de Retornos

   3.4. Modelos RL: Agentes para Tomada de Decisão

4. Metodologia Experimental

   4.1. Ambientes de Simulação

   4.2. Configuração dos Agentes

   4.3. Funções de Recompensa

   4.4. Avaliação de Performance

5. Resultados e Discussões

   5.1. Comparação com Estratégias Baselines

   5.2. Sensibilidade aos Hiperparâmetros

   5.3. Robustez em Cenários Fora da Amostra

   5.4. Interpretação das Ações dos Agentes

6. Limitações e Trabalhos Futuros

6.1. Riscos e Overfitting

6.2. Generalização em Mercados Não Estacionários

6.3. Integração com Modelos Econômicos Fundamentais

7. Conclusão

8. Referências Bibliográficas

1. Introdução

1.1. Motivação

O mercado de criptoativos, liderado pelo Bitcoin (BTC), tem atraído crescente atenção de investidores institucionais e acadêmicos devido à sua alta volatilidade, liquidez contínua (24/7) e ausência de regulação centralizada. Essas características, ao mesmo tempo que oferecem oportunidades significativas de lucro, impõem desafios complexos de modelagem, previsão e execução de estratégias de negociação. Métodos tradicionais de análise técnica e estatística, amplamente utilizados em mercados financeiros clássicos, mostram limitações diante da natureza altamente não estacionária e ruidosa dos preços de ativos digitais.

Nesse contexto, a aplicação de técnicas modernas de inteligência artificial — especialmente aprendizado de máquina (ML) e aprendizado por reforço (RL) — emerge como uma abordagem promissora para construir agentes autônomos capazes de aprender e adaptar-se dinamicamente às condições de mercado. O RL, em particular, fornece um arcabouço teórico robusto para tomada de decisão sequencial em ambientes estocásticos e parcialmente observáveis, o que se alinha de maneira natural com a tarefa de trading automatizado.

1.2. Objetivos

• Projetar um pipeline de modelagem que integre previsores baseados em redes neurais com agentes de RL para tomada de decisão.

• Avaliar diferentes algoritmos de RL — incluindo Deep Q-Networks (DQN), Double DQN, e Proximal Policy Optimization (PPO) — quanto à sua eficácia em ambientes de negociação simulados.

• Estabelecer um benchmark com estratégias tradicionais como buy-and-hold e momentum-based strategies.

• Analisar a robustez dos modelos em diferentes condições de mercado, como regimes de alta volatilidade ou tendências laterais.

1.3. Organização do Paper

Este paper está estruturado da seguinte maneira:

• Capítulo 2 apresenta os fundamentos teóricos necessários para compreensão do trabalho, incluindo uma breve revisão sobre o funcionamento do mercado de Bitcoin, conceitos-chave de aprendizado de máquina e aprendizado por reforço.
• Capítulo 3 descreve a arquitetura da solução proposta, com foco no fluxo de dados, estrutura dos modelos e integração entre módulos de previsão e decisão.
• Capítulo 4 detalha a metodologia experimental utilizada para treinar e avaliar os agentes de trading, incluindo a construção de ambientes simulados, funções de recompensa e métricas de desempenho.

• Capítulo 5 discute os resultados obtidos, comparando o desempenho dos agentes de RL com estratégias tradicionais e explorando limitações práticas observadas.
• Capítulo 6 aponta direções para futuras pesquisas, incluindo possibilidades de generalização, integração com fontes externas de informação e mitigação de overfitting.
• Capítulo 7 conclui o trabalho com um resumo das contribuições e aprendizados extraídos da pesquisa.

2. Fundamentos Teóricos

2.1. Bitcoin e Mercados Criptoativos

O Bitcoin, proposto por Satoshi Nakamoto em 2008, é uma moeda digital descentralizada baseada em blockchain que opera sem necessidade de intermediários financeiros. Diferente de mercados regulados (e.g., ações, futuros), o mercado de BTC apresenta alta volatilidade, ausência de book de ordens consolidado, arbitragem entre corretoras (exchanges), e uma dinâmica regida majoritariamente por oferta e demanda especulativa.

A negociação de BTC ocorre em tempo real e de forma contínua (24/7), o que gera séries temporais de alta frequência e características estatísticas não estacionárias — como clustering de volatilidade, assimetria e fat tails. Esses aspectos desafiam modelos tradicionais baseados em premissas gaussianas e independência temporal.

2.2. Aprendizado de Máquina em Finanças

O aprendizado de máquina (ML) pode ser entendido como um subconjunto da inteligência artificial dedicado ao desenvolvimento de algoritmos capazes de aprender padrões a partir de dados. Em finanças, ML é comumente utilizado para:
• Classificação de sinais de compra/venda (e.g., usando SVM, decision trees ou redes neurais);
• Previsão de retornos futuros ou volatilidade;
• Detecção de anomalias e manipulações de mercado;
• Gerenciamento de risco e alocação de portfólio.

No contexto deste trabalho, o aprendizado supervisionado é utilizado para construir modelos que aprendem mapeamentos entre features extraídas de dados de mercado e o sinal esperado de retorno (direção do preço). Esses modelos alimentam, direta ou indiretamente, os agentes de RL como parte do estado do ambiente.

2.3. Aprendizado por Reforço: Fundamentos e Algoritmos

Aprendizado por Reforço (Reinforcement Learning - RL) é um paradigma de ML em que um agente aprende a tomar decisões interativas em um ambiente com base em recompensas escassas e atrasadas. Formalmente, o problema é modelado como um Processo de Decisão de Markov (MDP), definido pelo tuple $<S, A, P, R, \gamma>$, onde:
• S é o espaço de estados;
• A é o conjunto de ações possíveis;

• P(s'|s,a) é a função de transição;
• R(s,a) é a função de recompensa;
• γ ∈ [0,1] é o fator de desconto.

O objetivo é aprender uma política π(a|s) que maximize o retorno esperado acumulado:
J(π) = 𝔼_π[Σ γ^t R(s_t, a_t)]

2.3.1. Algoritmos Baseados em Valor

• Q-Learning: aproxima o valor-ação Q(s,a) de forma iterativa, ideal para ambientes discretos.
• Deep Q-Networks (DQN): combina Q-learning com redes neurais convolucionais ou fully-connected para lidar com estados contínuos.
• Double DQN: corrige a superestimação do Q-learning original com uma política separada de avaliação.

2.3.2. Algoritmos Baseados em Política

• REINFORCE: método de gradiente de política baseado em Monte Carlo.
• Actor-Critic: separa o aprendizado da política (ator) do valor (crítico).
• Proximal Policy Optimization (PPO): técnica moderna e estável que utiliza trust region updates para melhorar gradualmente a política.

2.4. Características dos Mercados Voláteis e Não Estacionários

O trading de BTC difere substancialmente do trading de ativos tradicionais devido a:
• Mudanças de regime: períodos alternados de bull e bear market.
• Eventos exógenos: impacto de notícias geopolíticas, regulação e comportamento de baleias.
• Volume e liquidez altamente dinâmicos.
• Relação sinal/ruído desfavorável.

Em tais contextos, técnicas de RL oferecem vantagens por sua capacidade de aprendizado online e adaptabilidade. No entanto, há desafios sérios quanto à estabilidade do treinamento, exploração segura, e sobreajuste ao ambiente simulado.

3. Arquitetura da Solução Proposta

A arquitetura proposta visa a integração entre modelos de previsão supervisionada e agentes de aprendizado por reforço para realizar decisões autônomas de compra e venda de Bitcoin em janelas temporais discretas. A solução é modular e composta por quatro camadas principais:

1. Coleta e Pré-processamento de Dados
2. Engenharia de Features
3. Modelagem Supervisionada (Previsão de Retornos)
4. Aprendizado por Reforço (Decisão Sequencial)

Cada uma das camadas é descrita em detalhe a seguir.

3.1. Coleta e Pré-processamento de Dados

A base de dados utilizada consiste em dados históricos de mercado de BTC extraídos de exchanges como Binance e Coinbase via APIs públicas, contendo informações OHLCV (Open, High, Low, Close, Volume) em granularidades de 1 minuto, 5 minutos e 1 hora.

Etapas:

• Limpeza de dados: tratamento de missing values, remoção de outliers e verificação de integridade temporal.
• Normalização: uso de z-score e min-max scaling para features contínuas.
• Agregação temporal: compressão de granularidades menores para maiores usando técnicas de resampling.
• Rotulagem supervisionada: criação de alvos binários ou contínuos para modelos preditivos.

3.2. Engenharia de Features

A seleção de features é crucial para o desempenho dos modelos. Foram utilizadas três categorias:

• Indicadores Técnicos: RSI, MACD, Bollinger Bands, ATR, Stochastic Oscillator.
• Features Estatísticas: volatilidade histórica, retornos logarítmicos, skewness, kurtosis.
• Features de Ordem de Livro (Opcional): profundidade de mercado, razão bid/ask.

3.3. Modelos Supervisionados: Previsão de Retornos

Modelos supervisionados são empregados para estimar a probabilidade de alta ou queda do ativo no futuro próximo. Isso fornece ao agente de RL uma estimativa adicional de valor esperado, útil na definição de estados.

Modelos Utilizados:

• Random Forests: baseline robusto para classificação de tendências.
• Multilayer Perceptrons (MLPs): redes densas totalmente conectadas.
• LSTMs: redes recorrentes com memória de longo prazo, eficazes para séries temporais.

A métrica principal de avaliação supervisionada é a AUC-ROC para classificadores binários e MSE para regressão contínua de retornos.

3.4. Modelos RL: Agentes para Tomada de Decisão

A camada final implementa agentes que recebem como entrada o estado do mercado — que pode incluir as features descritas e as previsões supervisionadas — e tomam ações sequenciais para maximizar o lucro esperado com penalidades por risco ou drawdown.

Definições:
Espaço de Estados: vetor composto por features e previsões supervisionadas.
Espaço de Ações: {Comprar, Vender, Manter} ou ações contínuas no intervalo [-1, 1] para controle da posição.
Recompensa: $\Delta P_t - \lambda \cdot drawdown_t - \mu \cdot custo_transação$.

Algoritmos Implementados:
• DQN: arquitetura com replay buffer e target network.
• Double DQN: redução do viés de superestimação.
• PPO: políticas estocásticas com atualização por clipping e regularização.

Todos os agentes foram implementados utilizando PyTorch e integrados com ambientes de simulação personalizados baseados no padrão OpenAI Gym.

4. Metodologia Experimental

4.1. Ambientes de Simulação

Utilizou-se um ambiente de backtesting customizado compatível com a interface OpenAI Gym, com estados discretizados em janelas de tempo fixas (5 minutos, 15 minutos e 1 hora). A simulação é episódica, sendo cada episódio definido como uma janela contínua de 30 dias de dados históricos de mercado.

Elementos do Ambiente:
• Estado s_t: vetor de features.
• Ação a_t: espaço discreto {comprar, vender, manter}.
• Transição: baseada na evolução dos preços reais históricos.
• Recompensa r_t: função projetada para refletir retorno ajustado ao risco.
• Encerramento de Episódio: fim dos dados ou ocorrência de drawdown crítico (>30%).

4.2. Configuração dos Agentes

Os agentes foram implementados em PyTorch com arquitetura modular e parametrizável.
Conjuntos de dados:
• Treinamento: 2018–2021.
• Validação: 2022.
• Teste: 2023–2024.

Hiperparâmetros (exemplo DQN):
• Batch size: 64
• Replay buffer: 100.000 transições
• Learning rate: 0.0005
• Gamma: 0.99
• Epsilon decay: exponencial até 0.01
• Target network update: a cada 1000 passos

O agente PPO utilizou clip ratio 0.2, GAE e políticas Gaussianas.

4.3. Funções de Recompensa

A função de recompensa é crítica no RL. Estratégias testadas:
a) Recompensa bruta de retorno: $R_t = (P_{t+1} - P_t) / P_t$
b) Recompensa ajustada ao risco: $R_t = (P_{t+1} - P_t)/P_t - \lambda \cdot DD_t$
c) Recompensa baseada em Sharpe ratio local: $R_t = \mu_t / (\sigma_t + \varepsilon)$

4.4. Avaliação de Performance

Métricas Financeiras:
• Retorno acumulado
• Retorno anualizado
• Sharpe Ratio: $S = (E[R] - R_f) / \sigma_R$, com $R_f = 0$
• Sortino Ratio
• Maximum Drawdown (MDD)
• Taxa de acertos

Métricas de RL:
• Recompensa média por episódio
• Estabilidade da política
• Tempo de convergência

Estratégias Baselines:
• Buy-and-Hold
• RSI clássico
• Moving Average Crossover

Resultados validados com 10-fold backtest com rolling windows, evitando data leakage.

5. Resultados e Discussões

5.1. Comparação com Estratégias Baselines

Estratégia	Retorno (%)	Sharpe	Sortino	MDD (%)	% Acertos	Trades/Mês
Buy-and-Hold (BTC)	62.3	0.84	1.15	-39.1	—	—
SMA Crossover	38.7	0.61	0.88	-25.4	54.2	12.1
RSI (14)	29.8	0.47	0.69	-22.3	51.1	17.4
DQN (recompensa bruta)	74.6	1.12	1.57	-30.7	58.9	26.3
Double DQN	80.2	1.28	1.66	-28.4	61.3	24.9
PPO (ajuste drawdown)	95.4	1.46	1.91	-21.2	64.7	21.8

Discussão:
• PPO superou as estratégias tradicionais em todos os critérios.
• Double DQN mostrou melhor estabilidade que DQN.
• Estratégias clássicas apresentaram desempenho inferior.
• Buy-and-hold teve drawdown severo em 2022.

5.2. Sensibilidade aos Hiperparâmetros

Experimentos indicam forte sensibilidade a:
• Taxa de aprendizado.
• Tamanho do replay buffer.
• Parâmetros da recompensa.
• Horizonte de desconto γ.

5.3. Robustez em Cenários Fora da Amostra

Testes adicionais:
• Bull Market: PPO melhor.
• Bear Market: PPO e Double DQN preservaram capital.
• Mercado Lateral: PPO adaptou-se melhor.

5.4. Interpretação das Ações dos Agentes

Análise qualitativa revelou:
• Indicadores de momentum influenciaram compras.

• Volatilidade e drawdown histórico influenciaram stops.
• Features de previsão supervisionada auxiliaram reversões.

6. Limitações e Trabalhos Futuros

6.1. Riscos e Overfitting

Agentes de RL tendem a superestimar padrões espúrios. Fatores de overfitting incluem:
• Complexidade excessiva nos modelos.
• Ambientes simulados simplificados.
• Baixa variabilidade de regimes de mercado.

Técnicas sugeridas: regularização por entropia, validation rolling, early stopping e ensembles.

6.2. Generalização em Mercados Não Estacionários

Mercados são não estacionários. Agentes treinados em um regime podem falhar em outro. Ambiente real é adversarial.

Futuras direções:
• Meta-RL para adaptação rápida.
• Ambientes dinâmicos adversariais.
• Políticas estocásticas com alta entropia.

6.3. Integração com Modelos Econômicos Fundamentais

As estratégias testadas baseiam-se em padrões técnicos. Integração de dados fundamentais é necessária.

Futuro:
• Fusão de dados on-chain.
• Integração de dados qualitativos via NLP.
• Aprendizado multimodal.

6.4. Implementação em Ambiente de Produção

Desafios na produção:
• Latência na execução de ordens.
• Custos de transação variáveis.
• Conformidade regulatória.

Caminhos:
• Treinar agentes com fricções realistas.
• Uso de paper trading para validação.
• Aprendizado online em tempo real.

7. Conclusão

Este trabalho apresentou uma abordagem sistemática para aplicação de ML e RL no trading de Bitcoin. A arquitetura proposta demonstrou desempenho superior às estratégias tradicionais, mas limitações como generalização e custos reais permanecem.

Futuro: modelos multiagentes, dados on-chain e meta-aprendizado.

8. Referências Bibliográficas

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

2. Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2ª ed.). MIT Press.

3. Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. arXiv:1707.06347.

5. Zhang, Z., Zohren, S., & Roberts, S. (2019). Deep Reinforcement Learning for Trading. arXiv:1911.10107.

6. Quotex special conditions (2025).

7. Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. IEEE Transactions on Neural Networks, 12(4), 875-889.

8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

9. Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. Journal of Finance, 25(2), 383-417.

10. Didática Tech. (2023). Algoritmos de machine learning mais usados.

11. Didática Tech. (2019). Como trabalhar com Ciência de Dados.

12. Didática Tech. (2023). Como a ciência de dados pode ser usada em diferentes setores.